

# AENN: A GENERATIVE ADVERSARIAL NEURAL NETWORK FOR WEATHER RADAR ECHO EXTRAPOLATION

Jinrui Jing, Qian Li \*, Xinya Ding, Nengli Sun, Rong Tang, Yali Cai

College of Meteorology and Oceanography, National University of Defense Technology, 60 Shuanglong Road, Nanjing, China -  
(jingjinrui18, liqian, dingxinya, sunnengli, tangrong, caiyali)@nudt.edu.cn

Commission III, WG III/8

**KEY WORDS:** Weather Radar, Radar Echo Extrapolation, Deep Learning, Recurrent Neural Network, Generative Adversarial Network, Adversarial Training, Short-term Weather Forecasting

## ABSTRACT:

Weather radar echo is one of the fundamental data for meteorological workers to weather systems identification and classification. Through the technique of weather radar echo extrapolation, the future short-term weather conditions can be predicted and severe convection storms can be warned. However, traditional extrapolation methods cannot offer accurate enough extrapolation results since their modeling capacity is limited, the recent deep learning based methods make some progress but still remains a problem of blurry prediction when making deeper extrapolation, which may due to they choose the mean square error as their loss function and that will lead to losing echo details. To address this problem and make a more realistic and accurate extrapolation, we propose a deep learning model called Adversarial Extrapolation Neural Network (AENN), which is a Generative Adversarial Network (GAN) structure and consist of a conditional generator and two discriminators, echo-frame discriminator and echo-sequence discriminator. The generator and discriminators are trained alternately in an adversarial way to make the final extrapolation results be realistic and accurate. To evaluate the model, we conduct experiments on extrapolating 0.5h, 1h, and 1.5h imminent future echoes, the results show that our proposed AENN can achieve the expected effect and outperforms other models significantly, which has a powerful potential application value for short-term weather forecasting.

## 1. INTRODUCTION

The weather radar is one of the primary instruments for atmospheric remote sensing. Its detected data, as known as the weather radar echo, is used widely by forecasters for weather systems detection, diagnostic studies and short-term forecasting. Among the techniques based on radar data for short-term weather forecasting, the radar echo extrapolation, which is to predict the subsequent radar echoes based on the present and historical radar observations, is one of the most important. Accurate extrapolation could provide reliable early-warning for severe convective systems to the public and reduce the potential losses caused by disastrous weather, thus has long been valued and studied by researchers (Wilson et al., 1998).

Traditional extrapolation methods, including the centroid tracking (Dixon and Wiener, 1993), cross-correlation (Rinehart and Garvey, 1978) and optical flow (Woo and Wong, 2017), mainly rely on extrapolating the echo linearly with the calculated motion vectors, where the motion vectors for centroid tracking are cell-wise and for the other two are region-wise. However, their extrapolation capacity and application ability are both limited considering the following two issues. For the first, for calculating smoothing and relative effective motion vectors, it usually needs additional constraint conditions which are restricted to some specific situations or requires complicated parameter settings which are chosen mainly depends on experiences. For the second, the actual weather systems move and evolve in pretty complex patterns, such as

the rotation, formation and dissipation, which is not possible to well modeling them only by plain linear extrapolation.

Recently, the deep learning approaches for radar echo extrapolation and precipitation nowcasting have shown their significant progress. By training a sophisticated neural network architecture like Convolutional Neural Network (CNN) (Klein et al., 2015) or Recurrent Neural Network (RNN) (Shi et al., 2015) on a large-scale real-life radar echo dataset, a prediction model which has a powerful modeling capacity can be obtained and used for end-to-end echo extrapolation.

The success of the deep learning extrapolation methods is indeed promising for enhancing the nowcasting performance, but for them there still remains a problem of blurry prediction, that is, the predicted echo details are getting lost and the echo appearance is becoming fuzzy increasingly as the extrapolation goes further into the future, since they choose training objectives like Mean Square Error (MSE) or Mean Absolute Error (MAE) which would lead averaging all possible predictions and losing echo details.

In this paper, considering that the meteorologists actually are more desirable to aware of precise specifics of weather conditions at particular positions, our main target is to generate accurate and realistic echo extrapolation results. To achieve this, motivated by the recent success of Generative Adversarial Network (GAN) (Goodfellow et al., 2014) in generating realistic data distribution in the computer vision field, we propose an Adversarial Extrapolation Neural Network (AENN),

---

\* Corresponding author

which is composed of a conditional generator, an echo-frame discriminator and an echo-sequence discriminator. The conditional generator is used to generate the prediction of future echoes condition on input echo sequence, the echo-frame discriminator tries to distinguish each extrapolated echo frame from corresponding real echo frame, while the echo-sequence discriminator tries to distinguish the whole extrapolation sequence from the ground-truth echo sequence. The adversarial training is implemented between the conditional generator and two discriminators to prompt the extrapolation results to be more realistic and accurate.

For model evaluation, we conduct extrapolation experiments using the AENN to predict future echoes at 0.5, 1 and 1.5 hours based on the last half hour of echo observations. The qualitative and quantitative evaluation results both demonstrate the effectiveness of the AENN, indicates that it can be used to promote the operation of accurate short-term weather forecasting.

## 2. DATASET AND METHOD

### 2.1 Dataset

For model training and test, a real-life doppler weather radar echo dataset has been constructed in this paper. The radar echo base data we pre-processing is provided by the National Meteorological Information Center of China, which contains data collected by five CINRAD/SA doppler weather radars deployed in Hangzhou, Nanjing, Xiamen, Changsha and Fuzhou from 2016-2017. From the two years' data, we further select 175 rainy days' data to build our final dataset, for the reason that the rainy days' precipitation echo is our main extrapolation target.

For raw data pre-processing, firstly, it has been interpolated into the cartesian coordinate from the polar coordinate. The horizontal resolution of the interpolation grid is 1km while the vertical resolution is 0.5km. The interpolated Constant Altitude Plan Position Indicator (CAPPI) echo images on 2.5km altitude are chosen in our work and the central 480x480 area is cropped and remained. Then, to be suitable for deep learning neural network model as input, the final resolution of the echo images has been resized to 256x256 by bilinear interpolation. In addition, for data normalization, the value of the echo reflectivity is clipped to be between 0 and 75dBZ firstly and then converted into gray-level which is between 0 and 1.

In this paper, we conduct experiments on extrapolating 3 echo images (0.5, 1 and 1.5 hours) based on the last half hours' consecutive echo images as input, since the detection interval of the CINRAD/SA radar is 6 minutes, which means that the length of the input echo sequence is 5 and the total length of both the input and prediction is 8. Therefore, each rainy day's echo images are divided into sequences of length 8 in a sliding way with a stride of 5. Totally, 3640 sequences are acquired and randomly split into a training set of 2552 sequences, a validation set of 360 sequences and a test set of 728 sequences.

### 2.2 Model

Our model AENN is a GAN structure and composed of a conditional generator, an echo-frame discriminator and an echo-sequence discriminator, as illustrated in Figure 1.

The conditional generator generates the extrapolation results  $\hat{\chi}_{0.5}$ ,  $\hat{\chi}_1$  and  $\hat{\chi}_{1.5}$  condition on the input echo sequence  $\chi_{t-4:t}$ , where 0.5, 1 and 1.5 denote the extrapolation time step and  $t-4:t$  denote the input time step,  $\chi \in \mathbb{R}^{256 \times 256}$ . It consists of an encoder, Convolution Long Short-Term Memory Network

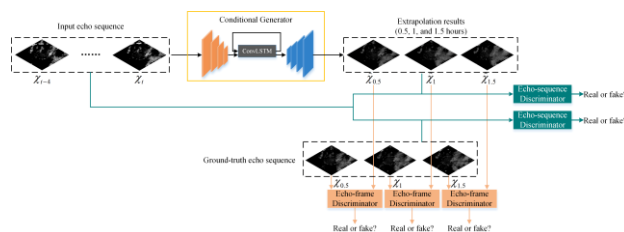


Figure 1. The architecture of the AENN.

(ConvLSTM) (Shi et al., 2015) and a decoder, the structure details will be described in Section 2.3.

The echo-frame discriminator is used to distinguish each generated extrapolation echo frame  $\hat{\chi}_{0.5}$ ,  $\hat{\chi}_1$  and  $\hat{\chi}_{1.5}$  from the real ground-truth ones  $\chi_{0.5}$ ,  $\chi_1$  and  $\chi_{1.5}$ , and is trained to try to judge the generated echo frame as the fake while the ground-truth as real. It is implemented with successive convolutional layers and one fully-connected layer to calculate a single scalar indicates the probability of the input would be real (1) or fake (0), see structure details in Section 2.3.

For the echo-sequence discriminator, it is trained to judge the authenticity of the whole extrapolation echo sequence  $\hat{\chi}_{0.5,1,1.5}$  or the whole ground-truth echo sequence  $\chi_{0.5,1,1.5}$ , trying to determine the former as fake while the latter as real. It could assure the temporal consistency of the whole extrapolation sequence, which is complementary to the role of the echo-frame discriminator. We implement the echo-sequence discriminator using the same neural network structure as the echo-frame discriminator, except that they are different in the input (sequence to frame).

The conditional generator, echo-frame discriminator and echo-sequence discriminator are trained competitively and alternately. The ideal nash equilibrium between them is that the generator could generate real enough extrapolation echo so that the discriminator could not distinguish them from reality. To achieve this desired target, we give our adversarial training details including the training objectives and strategies in Section 2.4.

### 2.3 Neural Network Structure

The conditional generator is consist of an encoder, ConvLSTM and a decoder, as shown in Figure 2.

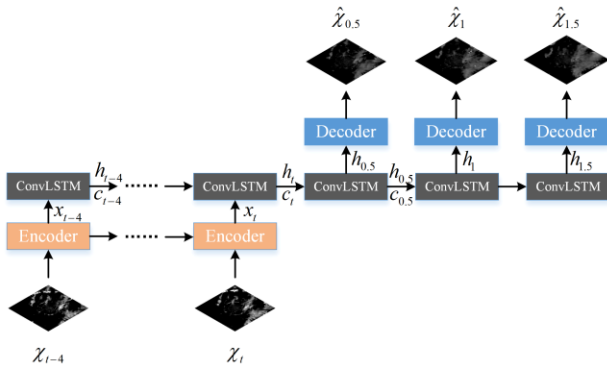


Figure 2. The structure of the conditional generator.

Firstly, the encoder maps the input echo sequence  $\chi_{t-4:t}$  from image space into feature space  $x_{t-4:t}$ , which encoding the spatial feature of echo components. Thus we implement it with the CNN specialized in spatial feature extracting. Specifically, it is consist of 3 convolutional layers with 32, 64 and 128 feature maps respectively. The first convolutional layer uses 5x5 size convolution kernel while the other two use 3x3 kernel, all of them uses 2x2 convolution stride. The activation function followed by each layer is the Rectified Linear Unit (ReLU) (Nair and Hinton, 2010).

Then, in the feature space, the ConvLSTM is used to modeling the temporal dynamics of the input echo sequence and extrapolate the future based on the dynamics. It works by recurrently updating its internal memory cell  $c$  and hidden state  $h$  according to the input  $x$  and historical information and controlled by three sigmoid gates  $i_t$ ,  $f_t$  and  $o_t$ . The updating equations are given as follows:

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + b_f) \quad (2)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o) \quad (4)$$

$$h_t = o_t \circ \tanh(c_t) \quad (5)$$

where  $\sigma$  is sigmoid activation function,  $*$  and  $\circ$  denote the convolutional operator and the Hadamard product respectively. Input  $x_t$ , memory cell  $c_t$ , hidden state  $h_t$ , input gate  $i_t$ , forget gate  $f_t$  and output gate  $o_t$  are both 3D tensors, where  $i_t$  controls the addition of the new information,  $f_t$  decides what previous memory will be forgotten and  $o_t$  regulates what information will be output. Weights  $W$  and biases  $b$  are both learning parameters of the ConvLSTM.

In this paper, we choose stack 2 ConvLSTM layers to increase the recurrence depth and modeling capability. Both of them have 128 feature maps and use 3x3 size kernel with 1x1 stride.

For the decoder, it transforms the state outs of the ConvLSTM  $h_{0.5,1,1.5}$  from feature space back to the image space  $\hat{\chi}_{0.5,1,1.5}$ . We implement it with a reversed structure compared to the encoder, consist of 3 deconvolutional layers with 3x3 size deconvolution kernel and 64, 32, 1 feature maps. The stride of the deconvolution is 2x2 to upsampling the feature map. All deconvolutional layers are also activated by the ReLU.

The echo-frame and echo-sequence discriminator share the same network structure, the only difference is that the echo-frame discriminator receives the echo frame ( $\hat{\chi}_n$  or  $\chi_n$ ,  $n \in \{0.5, 1, 1.5\}$ ) as input and the echo-sequence discriminator takes the real-echo sequence ( $[\chi_{t-4:t}, \chi_{0.5,1,1.5}]$ ) or fake-echo sequence as input ( $[\chi_{t-4:t}, \hat{\chi}_{0.5,1,1.5}]$ ), where  $[\cdot]$  denotes the concatenation in the feature dimension. They both output a single probability scalar. The network structure is shown in Figure 3.

The discriminator structure is comprised of five convolutional layers and one fully-connected layer. The five convolutional layers have 32, 64, 128, 256 and 512 feature maps respectively, where the first layer uses 5x5 size kernel with 2x2 stride and the others adopt 3x3 kernel with 2x2 stride. The average pooling is applied to the feature maps of the fifth convolutional layer and

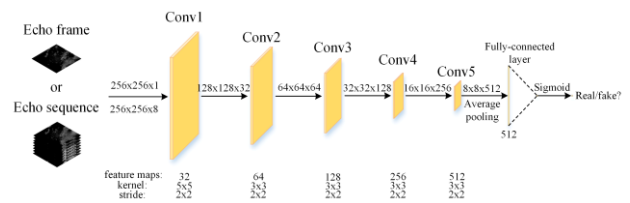


Figure 3. The structure of the discriminator.

the sigmoid is applied at last to convert the outputs of the fully-connected layer to the probability scalar. The activation function used in the discriminator is the Leaky ReLU (Maas et al., 2013) with a leaky rate of 0.2.

#### 2.4 Adversarial Training

We carry out the adversarial training between the conditional generator and two discriminators, which is alternately training one of them one step while keeping the parameters of the other one fixed. For training the conditional generator, our first objective is to make the extrapolation results close to the ground-truth and minimize the reconstruction loss, and the second goal is to fool the two discriminators judging the generated extrapolation results as real (1). Therefore, the loss function of the conditional generator  $\ell_g$  can be defined as:

$$\ell_g = \lambda_1 \ell_{rec} + \lambda_2 \ell_{seq}^{adv-g} + \lambda_3 \ell_{fra}^{adv-g} \quad (6)$$

where  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are weights corresponding to the reconstruction loss  $\ell_{rec}$ , echo-sequence adversarial loss  $\ell_{seq}^{adv-g}$  and echo-frame adversarial loss  $\ell_{fra}^{adv-g}$  respectively, they are set as hyper-parameter. For reconstruction loss  $\ell_{rec}$ , we adopt the sum of the MSE and MAE. For the two adversarial losses  $\ell_{seq}^{adv-g}$  and  $\ell_{fra}^{adv-g}$ , they are formulated by:

$$\ell_{seq}^{adv-g} = \ell_{bce} \left( d_{seq} \left( [\chi_{t-4:t}, \hat{\chi}_{0.5,1,1.5}] \right), 1 \right) \quad (7)$$

$$\ell_{fra}^{adv-g} = \sum_n^{\{0.5,1,1.5\}} \ell_{bce} \left( d_{fra} \left( \hat{\chi}_n \right), 1 \right) \quad (8)$$

where  $d_{seq}$  denotes the echo-sequence discriminator and  $d_{fra}$  denotes the echo-frame discriminator.  $\ell_{bce}$  is the binary cross-entropy loss function formulated by:

$$\ell_{bce}(\hat{p}, p) = -p \log(\hat{p}) - (1-p) \log(1-\hat{p}) \quad (9)$$

where  $p$  is the label (0 or 1) and  $\hat{p}$  is the logit (between 0 and 1).

When training the two discriminators, the only objective is to make them correctly judge the ground-truth label as real (1) and the extrapolation echoes generated by the generator as fake (0). Thus the loss function of the echo-frame discriminator  $\ell_{fra}^{adv-d}$  and echo-sequence discriminator  $\ell_{seq}^{adv-d}$  are defined as follows:

$$\ell_{fra}^{adv-d} = \sum_n^{\{0.5, 1, 1.5\}} \ell_{bce}(d_{fra}(\chi_n), 1) + \ell_{bce}(d_{fra}(\hat{\chi}_n), 0) \quad (10)$$

$$\ell_{seq}^{adv-d} = \ell_{bce}(d_{seq}([\chi_{t-4t}, \chi_{0.5, 1, 1.5}]), 1) + \ell_{bce}(d_{seq}([\chi_{t-4t}, \hat{\chi}_{0.5, 1, 1.5}]), 0) \quad (11)$$

In addition to the training objectives, considering that the number of parameters and model complexity for the generator and discriminator are different, and thus their convergence rate is also inconsistent, in our adversarial training strategy, we training the generator and discriminator with different updating rates. To be specific, in this paper, the updating ratio of the conditional generator to the discriminators is 3:1, which means that the generator will be updated 3 times every updating step of the discriminators.

### 3. EXPERIMENTS

#### 3.1 Experimental Settings

We implement our model using Python and Tensorflow and conduct the experiments on 4 RTX 2018Ti GPUS. For model parameters initialization, we initialize all the kernel weights using Xavier initializer (Glorot and Bengio, 2010) and all the biases to 0. During the model training stage, the weights corresponding to the reconstruction loss  $\lambda_1$ , echo-sequence adversarial loss  $\lambda_2$  and echo-frame adversarial loss  $\lambda_3$  are set to 1, 0.003 and 0.003 respectively, which assures both of the sub-loss poses the balanced effect on the training process. Both of the conditional generator and two discriminators are trained by Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.0001.

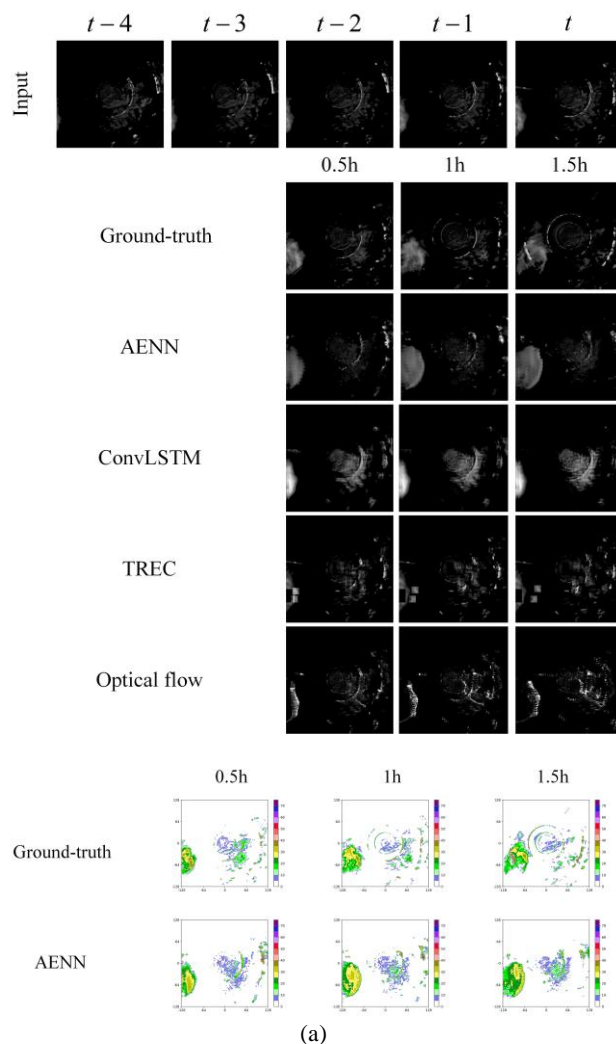
We compare our model AENN with two traditional extrapolation methods, Tracking Radar Echoes by Correlation (TREC) and Optical flow, and one state-of-the-art deep learning extrapolation model, ConvLSTM. We first give several extrapolation samples predicted by these models to qualitatively evaluate the performance of the models, which will be described in Section 3.2. Then the quantitate evaluation experiment is carried out on four commonly used precipitation nowcasting skill metrics, Probability of Detection (POD), False Alarm Rate (FAR), Critical Success Index (CSI) and Heidke Skill Score (HSS), the evaluation results are given in Section 3.3.

#### 3.2 Extrapolation Samples

To qualitative analyze the effectiveness of our model, we let the AENN and other three comparison models to extrapolate three echo evolution examples including the evolutionary process of a cell echo, a layered echo and a typhoon echo, the extrapolation results are shown in Figure 4.

In the ground-truth of Figure 4 (a), the cell echo at Nanjing, China 6 Sep 2016 09:56 UTC is appearing and moving from west to the east. It can be seen that our model AENN successfully models the echo motion and predicts a basically correct echo shape. The ConvLSTM does not well predict the echo motion and even amplifies the central clutter. The TREC and Optical flow perform worst, their extrapolated echo is much disheveled than others since it's hard for them to obtain a motion vector field with high spatial consistency.

As illustrated in Figure 4 (b), the ground-truth is that a layered echo at Hangzhou, China 14 Sep 2016 13:56 UTC is gradually developing and moving towards the north. The TREC and Optical flow are unable to effectively maintain the integral echo shape and not to mention predicting the echo evolution. For the





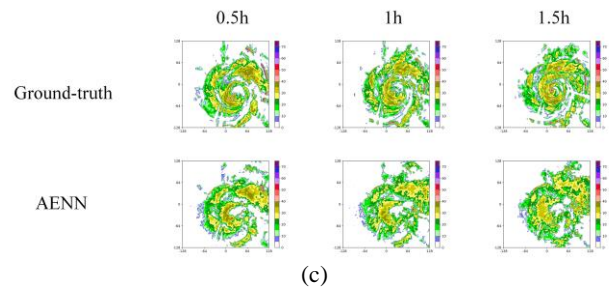
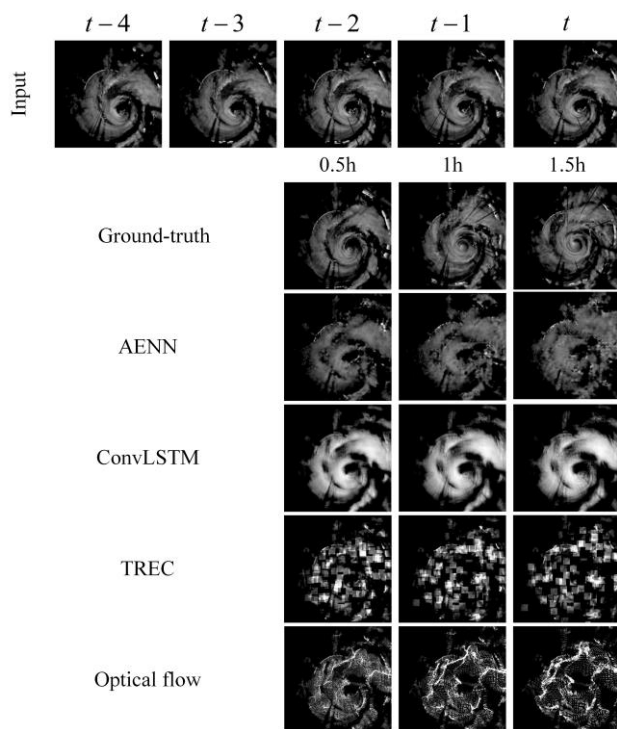
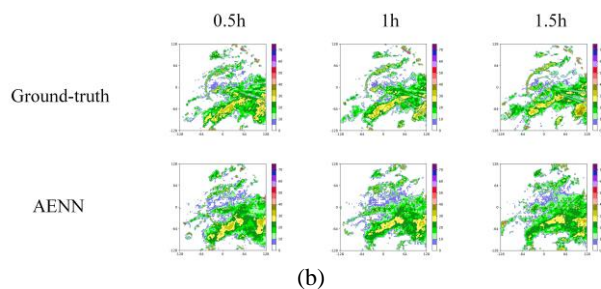
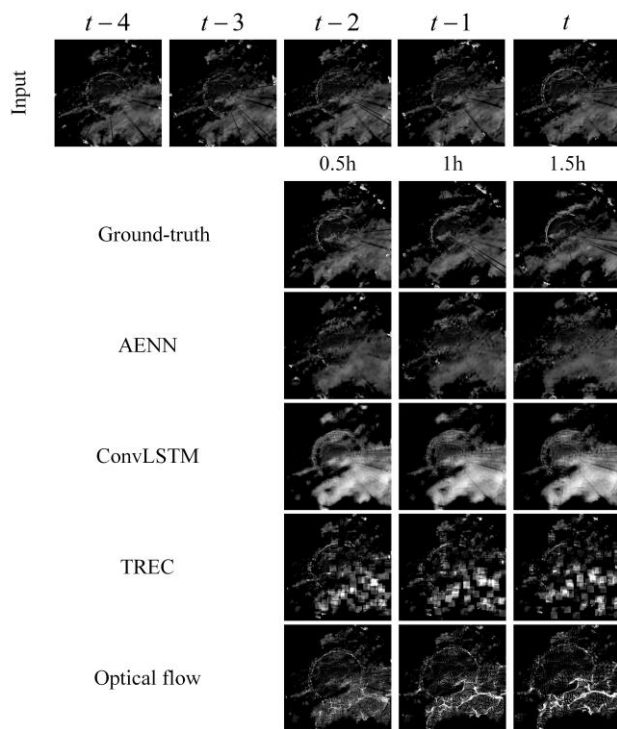


Figure 4. Three extrapolation samples by the AENN, ConvLSTM, TREC and Optical flow. (a) A cell echo at Nanjing, China 6 Sep 2016 09:56 UTC; (b) A layered echo at Hangzhou, China 14 Sep 2016 13:56 UTC; (c) The “Meranti” echo at Xiamen, China 14 Sep 2016 15:16 UTC.

ConvLSTM, its extrapolation result shows an obvious blurry problem and inconspicuous motion. While for our AENN, it not only predicts the echoes which contain more internal texture details but also modeling the echo evolution accurately.

Figure 4 (c) shows a typhoon echo called “Meranti” at Xiamen, China 14 Sep 2016 15:16 UTC. It has a more complicated structure and temporal dynamic than the cell echo and layered echo in Figure (a) and (b). Compared to the TREC and Optical flow, the two deep learning models ConLSTM and AENN have the ability to roughly generate the future shape of the “Meranti”. But for ConvLSTM, its modeling capability is limited that the echo dynamics have not been well modeled. For our AENN, although its prediction performance is also getting worse as forecasting timing increasing, it has considered and predicted both the echo shape and temporal dynamics.

Overall, our AENN achieves the best performance among the four models. The blurry problem has been alleviated and the extrapolated echoes are more accurate and realistic, which would attribute to the adversarial training. In addition, the AENN also displays the merit that it can model the echo motion and evolution, shows that it can automatically learn a valuable representation of echo dynamics which will contribute to the advances of research in radar meteorology.

### 3.3 Evaluation Results

In this section, we evaluate the performance of the models quantitatively on four evaluation metrics, POD, FAR, CSI and HSS. To calculate them, we first convert the pixel-level values of the extrapolated echo back into reflectivity factor values and then obtain the rainfall rate using Z-R relationship as:

$$Z = 10 \log a + 10b \log R \quad (12)$$

where  $Z$  is the reflectivity factor value and  $R$  is the rainfall rate,  $a$  and  $b$  are two constants set as 58.53 and 1.56.

The POD, FAR, CSI and HSS are finally calculated at a threshold of 0.5 (mm/h) rainfall rate by the following equations:

$$POD = \frac{n_h}{n_h + n_m} \quad (13)$$

$$FAR = \frac{n_f}{n_h + n_f} \quad (14)$$

$$CSI = \frac{n_h}{n_h + n_m + n_f} \quad (15)$$

$$HSS = 2 \frac{n_h n_c - n_m n_f}{(n_h + n_m)(n_m + n_c) + (n_h + n_f)(n_f + n_c)} \quad (16)$$

where  $n_h$  denotes the hits (ground-truth=1, prediction=1),  $n_m$  denotes the misses (ground-truth=1, prediction=0),  $n_f$  denotes the false alarms (ground-truth=0, prediction=1) and  $n_c$  denotes the correct rejections (ground-truth=0, prediction=0). The higher value of the POD, CSI and HSS and a lower value of FAR indicates the precipitation nowcasting quality if better.

The evaluation results of the four models on the whole test set are given in Table 1. From the table we can see, the AENN achieves the almost best scores, its POD, CSI and HSS are the highest while the FAR close to the lowest, which outperforms other models significantly. Thus the effectiveness of our AENN is demonstrated both qualitatively and quantitatively.

Models	Evaluation metrics			
	POD	FAR	CSI	HSS
AENN	<b>0.636</b>	0.363	<b>0.482</b>	<b>0.574</b>
ConvLSTM	0.502	<b>0.335</b>	0.416	0.502
TREC	0.434	0.466	0.322	0.390
Optical flow	0.410	0.493	0.304	0.364

Table 1. Evaluation results of the four models on four metrics. The averaging score of all extrapolation steps is given. The bold figures indicate the best score.

#### 4. CONCLUSIONS

In this paper, we study the weather radar echo extrapolation for accurate short-term weather forecasting. To offer accurate extrapolation results which are hard for traditional extrapolation methods to obtain, we propose a deep learning model called Adversarial Extrapolation Neural Network (AENN). It is a Generative Adversarial Network (GAN) structure and utilizing the adversarial training to avoid the blurry prediction problem. The experimental results show that the AENN can generate accurate and realistic extrapolation echoes and the echo motion and evolution have been also modeled, indicates that the AENN can be used for effectively assist the actual weather forecasting practice. In our future work, we will further investigate how to extrapolate future echoes that have a more accurate shape which can match the ground-truth perfectly. We believe that maybe utilizing some of the morphometric losses would be useful.

#### ACKNOWLEDGEMENTS

This research was funded by the National Natural Science Foundation of China (Grant No. 41305138), the China Postdoctoral Science Foundation (Grant No. 2017M621700) and the National Key Research and Development Program of China (Grant No.2018YFC1507604).

#### REFERENCES

Dixon, M., Wiener, G., 1993. TITAN: Thunderstorm Identification, Tracking, Analysis, and Nowcasting—A radar-based methodology. *J. Atmos. Oceanic Technol.* 10, 785–797.

Glort, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks, in: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Presented at the Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 249–256.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (Eds.), *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2672–2680.

Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs].

Klein, B., Wolf, L., Afek, Y., 2015. A dynamic convolutional layer for short range weather prediction. Presented at the *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4840–4848.

Maas, A.L., Hannun, A.Y., Ng, A.Y., 2013. Rectifier nonlinearities improve neural network acoustic models, in: ICML Workshop on Deep Learning for Audio, Speech and Language Processing.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines, in: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10. Omnipress, USA, 807–814.

Rinehart, R.E., Garvey, E.T., 1978. Three-dimensional storm motion detection by conventional weather radar. *Nature* 273, 287–289.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., WOO, W., 2015. Convolutional LSTM Network: A machine learning approach for precipitation nowcasting, in: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 802–810.

Wilson, J.W., Crook, N.A., Mueller, C.K., Sun, J., Dixon, M., 1998. Nowcasting thunderstorms: a status report. *Bull. Amer. Meteor. Soc.* 79, 2079–2100.

Woo, W., Wong, W., 2017. Operational application of optical flow techniques to radar-based rainfall nowcasting. *Atmosphere* 8, 48.